Beginners Guide to Contributing

About

You seem to like this Jenkins thing well enough, but you'd like to contribute back to the project, help improve things, etc. Welcome!
This document is intended to help introduce you to the Jenkins project, help you find something that you're interested in helping out with.

For all contributors

- A *great* introduction to the Jenkins project is our **Governance Document** which should help give you an idea of how the project is structured, our beliefs, goals, etc.
- Join us on the IRC Channel (`#jenkins` on `irc.freenode.net`), great for instant feedback on ideas, questions, etc.
- Create a Jenkins user account for our bug tracker/wiki/etc on this page.
- Participate in our (*usually*) bi-weekly project meetings on IRC, see our Governance Meeting Agenda page for more details.

Are you interested in writing code?

General

- Create an account on GitHub and follow the Jenkins repositories that are interesting to you: github.com/jenkinsci

Jenkins *core*

We generally call *Jenkins core* the jenkins project itself, who lives there on github. Creating your own fork and submitting a pull request is a great way to begin contributing your fixes.

How about Java code?

General

- Help review/comment on/test Pending Pull Requests that are having "trouble" either being integrated or closed out for various plugins and core.
    - Sometimes plugin developers move on and don't notice your pull request, so in those cases, it'd be great if people can make some noise on the developer mailing list.
    - Want to get more involved and commit directly to the repository? Just ask! We generally give away the push access very liberally. This process hasn't been fully automated yet, but the intention is that you just need to ask to get the push access.
- Bookmark both javadoc.jenkins.io and ci.jenkins.io
- Be sure that you look over this brief guideline on submitting a "good" pull request.
- Java requirements may differ depending on the component, see their documentation or pom.xml

Contributer Agreement

More information about the copyrights can be found here.

Commit Messages

Jenkins' github repositories are hooked up to the scm-issue-link daemon.

- To submit a change and have that reflected in a bug report use text like [JENKINS-*nnnn*] in the commit message.
- To submit a change that fixes an issue use [FIXED JENKINS-*nnnn*] (or …`FIX`…, …`FIXES`…) in the commit message.

Obviously replace *nnnn* with the issue number.

For more information see the scm-issue-link page.

Code Style

In the core, we try to loosely follow the Oracle Java Code Conventions. Plugins are up to each owner, and we respect whatever choice they make.

Code Review

For changes that you feel comfortable, we normally expect folks to just push straight in. This includes localization, which only few of us can review. When you do this, you just need to be prepared that someone might yank that out later.

For other times, people feel more comfortable going through a pull request or a patch in the issue tracker, so that they have someone else's ack to be integrated, especially for the first few changes. Never hurts to have a second, or third opinion.

In *core*

- Fork the main Jenkins repo on GitHub and get the "core" building on your local machine. The Building Jenkins page should help get you started.
- Participate in the "UI Enhancements" project by experimenting with the Jenkins UI, or providing feedback to existing experiments (this may or may not be a great beginner project from a code standpoint).

In plugins

- Adopt a Plugin which doesn't have or doesn't seem to have a maintainer (list in JIRA). Make sure you read this section on the subject from our Governance Document.
- Pick up an old enhancement or bug report from the issue tracker
- Start writing your own plugin, following our Plugin tutorial
- Write an automatic tool installer to auto-install a piece of software you find useful (such as the existing mongodb auto-installer). The Adding tool auto-installer page should help get you started.

How about Ruby code?

- Start developing a plugin to meet a specific need you have with the Ruby plugin support in Jenkins, check out this Getting started with Ruby plugins guide.
- Dig into the source code for existing Ruby plugins and either extend them or fix outstanding bugs.

How about Puppet?

- Participating in the management of the Jenkins project itself via the jenkins-infra project on GitHub.
    - Help write rspec-puppet tests for existing modules, or some other kind of testing that can help automate testing of manifests/modules
- Contributing fixes, extensions to the semi-official puppet-jenkins module on GitHub

Do you speak languages other than English?

- Contribute localizations to your native tongue via the Internationalization project.
- Contribute localizations from any page within Jenkins by using the Translation Assistance Plugin.
- Help verify and/or proof-read existing translations.

Are you not interested in writing code?

General

- Learn more about Jenkins itself
- Use it! Some of the best ideas come out of putting Jenkins to use in your existing workflow
- Keep notes of what felt over-complicated, weird or otherwise confusing when getting started. For bonus points, file enhancement requests in JIRA so we can get these pain-points fixed.

Like talking to people about technology?

- Give a talk about Jenkins at a local JUG or other group about using Jenkins (the fledgling Jenkins CIA Program may be of interest as well)
- Organize a meetup of local Jenkins users, future users and developers

Help other newbies

- Help answer questions on the mailing lists
- Help answer questions the IRC Channel if you have the time (#jenkins on irc.freenode.net)
- Help answer questions tagged on Stack Overflow

Like writing or documenting things?

- Share your solutions for "common" problems that you have faced. Blog posts/case studies/write-ups of Jenkins usage in a particular environment, with specific plugins that were used to solve a problem are very useful (check out the Meet Jenkins page for ideas).

- Help tidy up or modernize older wiki pages that are aimed at helping new comers, such as those linked from the "Use Jenkins" page.
- For some technology X, talk/publish/post about "Jenkins + X" and how to make the two work together
    - Example: Hardware design: Experiences using Jenkins for ASIC Development
    - Example: iOS Development: Continuous Deployment of iOS Apps with Jenkins and TestFlight

Like packaging software?

- Pick your favorite platform and contribute a new packaging script, or help maintain the existing scripts

Like breaking/testing software?

- Learn about JIRA and help triage or otherwise verify existing bug reports.
    - For JIRAs without reproduction steps, helping to nail down the exact steps and environment needed to reproduce the bug can help developers tremendously
    - For JIRAs with reproduction steps, verifying that the bug is fixed when a developer submits a patch can also be very helpful.
- Test upcoming release candidates of Jenkins itself via the RC channel. It may also be worth familiarizing yourself with the Release Process
- Help verify and test the LTS (Long Term Support) release candidates before they're finalized, more details can be found on the LTS RC Testing page.

Unit tests

As for any other code, you should have unit tests for your contribution. If you're e.g. want to contribute a bug fix, you should have a unit test which exposes the bug in the old code and confirms that it's fixed after your change.

Note:

- Your pull requests have a higher chance to be integrated quickly if it contains tests
- most existing Jenkins core tests are in the jenkins-test-harness module